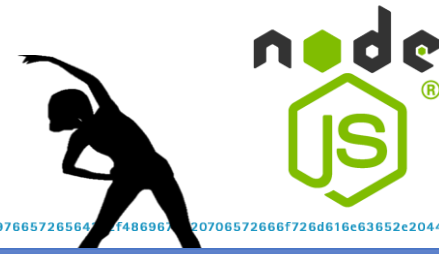


Atividade 1: Retângulo



- Alterar o **solucao-1.js**
- Obter, via teclado, os valores para cálculo do:
 - Perímetro do retângulo
 - Área do retângulo
- Utilize o módulo **readline**.
- Enviar arquivo compactado no formato **zip** para angela.de.a.ferreira@accenture.com
- Prazo: 23/10/2023

exemplo-readline.js

```
const readline = require("readline");

const readline = readline.createInterface({
  input: process.stdin,
  output: process.stdout,
});

readline.question("Qual o seu nome? ", (name) => {
  console.log(`Ola ${name}!`);
  readline.close();
});
```

Solucao-1.js

```
var rect = require('./retangulo-1.js')

function solveRect(l,b) {
  console.log("Solução para o retangulo com l = " + l + " e b =" + b);

  if (l < 0 || b < 0){
    console.log("as dimensoes do retangulo devem ser maior que zero: l= " +
l + ", e b+ " +b);
  }
  else{
    console.log("A area do retangulo com dimensoes comprimento = " + l + "
e largura = " + b + " e " + rect.area(l,b));
    console.log("O perimetro do retangulo com dimensoes comprimento = " + l
+ " e largura = " + b + " e " + rect.perimeter(l,b));
  }
}

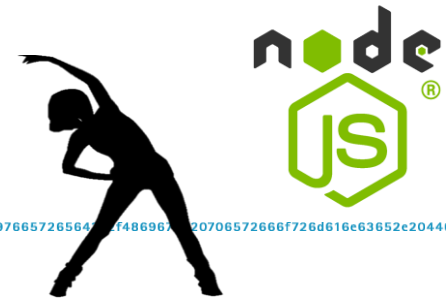
//solveRect(2,4);
//solveRect(3,5);
solveRect(-3,5);
```

Retangulo-1.js

```
exports.perimeter = function(x,y){
  return (2*(x+y));
},

exports.area = function(x,y){
  return(x*y)
}
```

Atividade 2: Calculadora

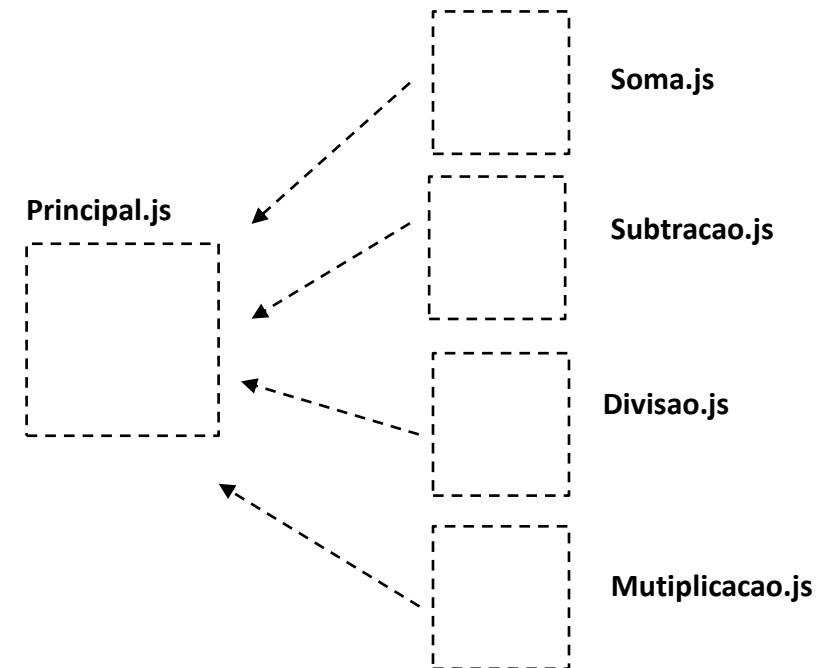


- Crie uma calculadora.
- Ela deve ter 5 arquivos.
- Sendo 4 com as operações básicas e 1 com a função principal.
- Usar o **prompt-sync**
- Enviar arquivo compactado (**sem a pasta node_module**) no formato **zip** para angela.de.a.ferreira@accenture.com
- Prazo: 23/10/2023

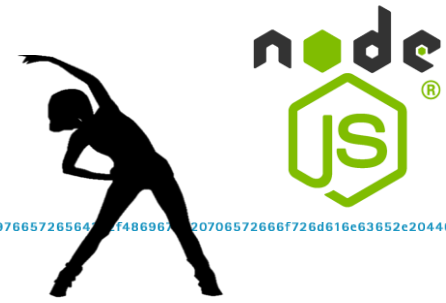
exemplo-promp-sync.js

```
const prompt = require('prompt-sync')();

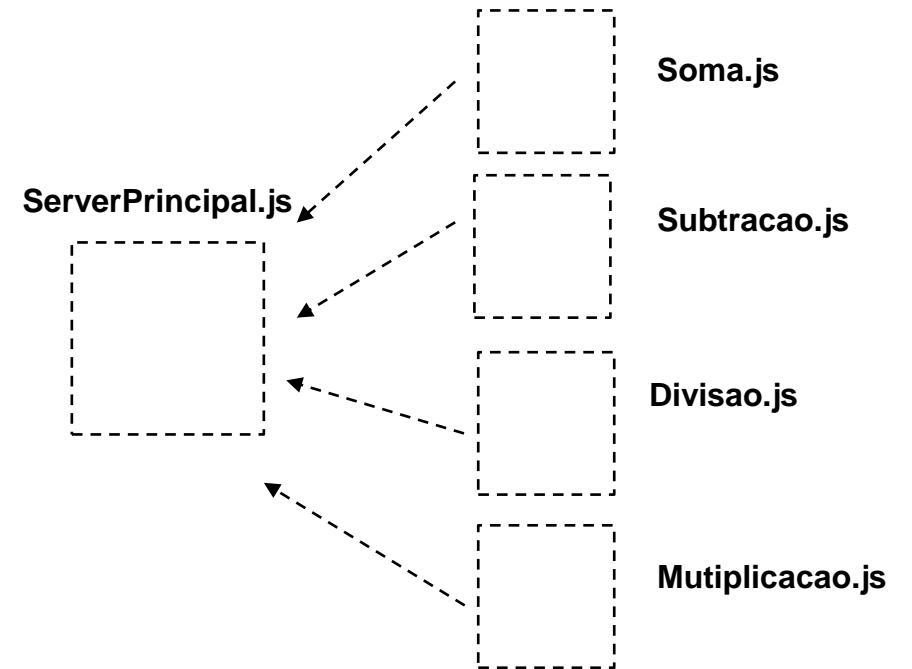
const num = prompt('Informe um número: ');
console.log('Seu número + 4 =');
console.log(Number(num) + 4);
```

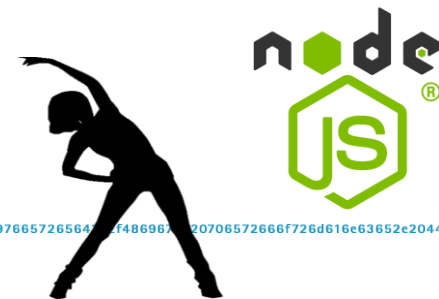


Atividade 3: API Calculadora



- Crie uma API Calculadora.
- A API deverá ter:
 - Pelo menos 4 end-points do tipo GET.
 - Usar os arquivos externos com as 4 operações básicas.
 - O Front-End deve enviar para o servidor Node os números e o tipo de operação que será realizada (opcional).
- Enviar arquivo compactado (**sem a pasta node_module**) no formato **zip** para angela.de.a.ferreira@accenture.com
- Prazo: 23/10/2023





Atividade 4: MySQL – Passagem de valor

- Criar uma tabela usuários no MySql.

```
CREATE TABLE `user` (  
  `id` smallint unsigned NOT NULL AUTO_INCREMENT,  
  `nome` varchar(200) NOT NULL,  
  `telefone` varchar(15) DEFAULT NULL,  
  `email` varchar(150) DEFAULT NULL,  
  `novidades` tinyint(1) NOT NULL,  
  `mensagem` text,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;
```

- Criar a API que vai receber os dados enviados do formulário.
- Salvar os dados na tabela.
 - PassagemDeValores.html
 - Node_Sparametro2.js
- Enviar arquivo compactado (**sem a pasta node_module**) no formato zip para angela.de.a.ferreira@accenture.com
- Prazo: 25/10/2023

Envie uma mensagem preenchendo o formulário abaixo

Seu Nome:

Seu Telefone:

Seu E-Mail:

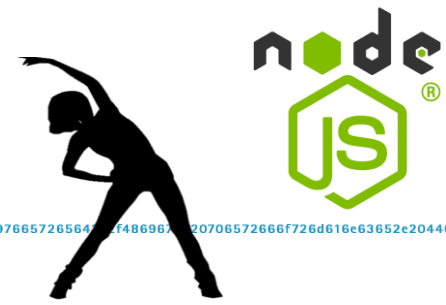
Deseja receber nossas novidades?

☒ Sim ☐ Não

Sua mensagem:

Enviar

Atividade 5: AXIOS ViaCEP



Crie uma API axios para consumir a API VIACEP.

- Criar uma pagina web simples para informar o CEP.
- Criar a tabela de CEP no mysql.
- Fazer GET na viacep.
- Recuperar os dados.
- Conectar no mySQL.
- Salvar em uma tabela o cep recuperado.
- <https://viacep.com.br/ws/01001000/json/>

```
CREATE TABLE IF NOT EXISTS `cep` (  
  `cep` VARCHAR(9) PRIMARY KEY,  
  `logradouro` VARCHAR(300) NOT NULL,  
  `complemento` VARCHAR(200) NULL,  
  `bairro` VARCHAR(200) NOT NULL,  
  `localidade` VARCHAR(200) NOT NULL,  
  `uf` VARCHAR(2) NOT NULL,  
  `ibge` VARCHAR(10) NOT NULL,  
  `gia` VARCHAR(200) NULL,  
  `ddd` VARCHAR(3) NOT NULL,  
  `siafi` VARCHAR(10) NOT NULL  
) ENGINE=InnoDB;
```